

네트워크 개론 Part 15

: TCP(Transmission Control Protocol)의 이해 3

글. 조인준 KBS 미디어기술연구소 차장

지난 편부터 TCP(Transmission Control Protocol)의 데이터 전송 관련 제어방식에 관한 토픽을 다루며, 현재 주로 사용되는 TCP의 전송 관련 제어방식이 만들어지기까지의 기술적 흐름에 대한 통찰을 드리고자 가장 단순한 제어방식에서 시작하여 각 방식의 한계를 극복하기 위해 개량된 방식을 차례차례 소개하는 형식으로 단순 송수신, Stop & Wait, Stop & Wait ARQ(Automatic Repeat ReQuest)에 관해 설명했습니다.



그림 1. Stop & Wait에서의 ACK 메시지와 패킷 전송

이상적인 상황에서만 가능한 단순 송수신은 송신 측 호스트에서 수신 측 호스트가 데이터를 수신할 수 있는 상태인지 아닌지 알 수가 없어서 수신 측이 수신 불가능한 상황에서도 데이터를 계속 보내는 문제가 생길 수 있습니다. 이를 해결하기 위한 Stop & Wait 방식은 [그림 1]과 같이 수신 호스트가 보내는 ACK 메시지로 수신 호스트의 준비상태를 체크하여 하나씩 패킷을 보내지만 네트워크 등의 문제로 ACK 메시지를 받지 못하면 무한히 ACK 메시지를 기다리며 더 이상의 패킷 송수신이 이루어지지 않는 문제가 발생할 수 있습니다. 그래서 ACK 메시지를 받기 위해 대기하는 시간의 타임아웃을 설정할 수 있는 Stop & Wait ARQ를 통해 무한대기에 빠지지 않으면서 전송에 실패한 데이터를 다시 보낼 방법이 고안되었으나, 이 방식 또한 Stop & Wait를 기반으로 하기에 ACK 메시지 한 번에 하나의 패킷만을 보내므로 네트워크 대역에 여유가 있더라도 더 많은 패킷을 더 빨리 보내지 못하는 답답한 데이터 전송 상황을 보일 수밖에 없었습니다.

언급한 요약과 같이 단순 송수신 → Stop & Wait → Stop & Wait ARQ로의 기술적 개선과정을 들여다보며 수신 측 호스트가 준비된 이후 데이터를 전송하고, ACK 메시지가 전송되지 않더라도 무한대기에 빠지지 않으면서 전송 오류를 바로잡기 위해 어떤 기술적 요소가 추가되었는지 이해할 수 있었습니다. 그리고 요약 마지막 부분의 Stop & Wait나 Stop & Wait ARQ의 전송속도 언급에서 전송속도를 높이는 방법을 이번 편에서 설명할 것 같은 예감이 독자 여러분께 밀려오고 있을 것 같습니다. 네, 그 느낌이 맞습니다.

이번 편에서는 전송속도를 높이기 위한 슬라이딩 윈도우(Sliding Window) 기법을 설명하고, 슬라이딩 윈도우를 사용한 현재 TCP의 주된 전송 관련 제어방식인 Go-Back-N ARQ와 Selective Repeat ARQ에 대한 설명을 다음 편에서 이어가겠습니다.

윈도우 사이즈 : 5

패킷 0 1 2 3 4 5 6 7 8 9

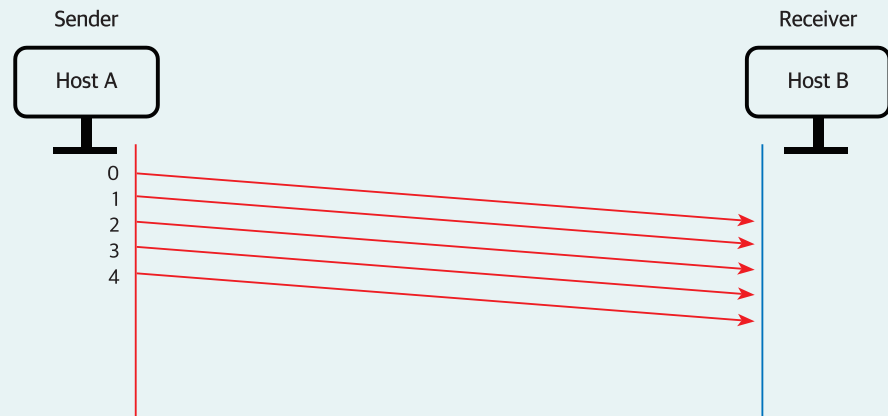


그림 2. 슬라이딩 윈도우 사이즈에 따른 패킷 전송 (최초 전송)

우선 슬라이딩 윈도우가 무엇인지 알아보겠습니다. 슬라이딩 윈도우는 [그림 1]과 같이 수신 측의 ACK를 받고 패킷을 하나씩 보내는 Stop & Wait 방식과는 달리 수신 측의 ACK가 없는 상황에서도 여러 개의 패킷을 보낼 수 있습니다. ACK 없이 보낼 수 있는 패킷의 최대 갯수는 TCP 헤더의 윈도우 사이즈(Window Size)로 정하며, 각 패킷은 TCP 헤더의 시퀀스 번호(Sequence Number)로 구분됩니다. [그림 2]는 0~9까지의 시퀀스 번호를 갖는 10개의 패킷을 윈도우 사이즈 5로 설정된 슬라이딩 윈도우로 Host A에서 Host B로 보낼 때 처음 5개 패킷이 전송된 상황을 나타냅니다. ACK 없이 보낼 수 있는 패킷의 수인 윈도우 사이즈가 5이므로 수신 측인 Host B의 ACK 없이도 Host A는 처음 5개의 패킷을 보낼 수 있습니다. Host A 위에 시퀀스 번호 0~9로 전송될 패킷을 나타내는 블록들과 함께 녹색 사각형으로 슬라이딩 윈도우를 표시했습니다. 슬라이딩 윈도우는 현재 윈도우 사이즈가 5이므로 Host A가 이미 송신한 5개의 패킷(시퀀스 번호 0~4)을 포함하고 있습니다. 붉은색 화살표는 Host A에서 Host B로 패킷이 전송된 것을 나타내고 화살표 시작점의 숫자(0, 1, 2, 3, 4)는 전송된 패킷의 시퀀스 번호입니다. Stop & Wait와 달리 수신 측의 ACK 없이도 시퀀스 번호 0~4까지 5개의 패킷이 연달아 전송되었지만 윈도우 사이즈와 같은 5개의 패킷이 ACK 없이 전송된 후 더 이상의 패킷 전송은 이루어지지 않습니다.

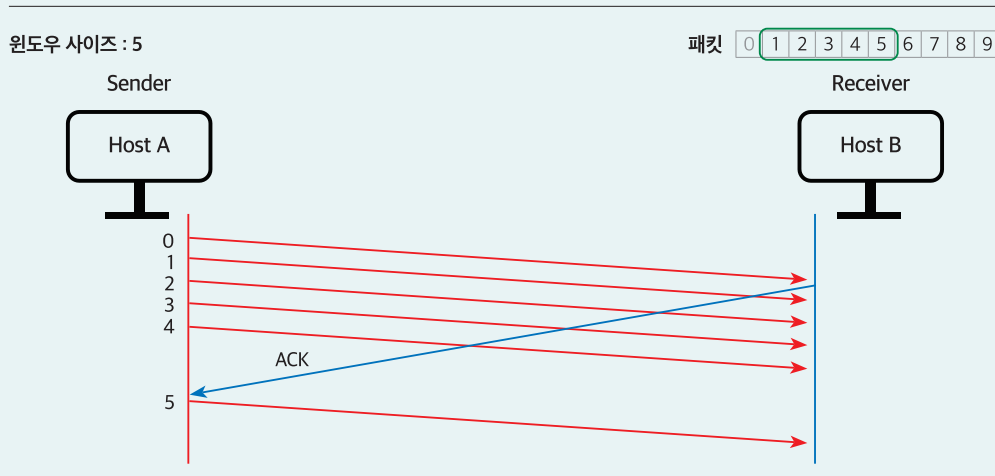


그림 3. 슬라이딩 윈도우 사이즈에 따른 패킷 전송 (ACK 메시지 수신 후)

원도우 사이즈와 같은 5개의 패킷이 이미 전송되어 더 이상의 전송이 멈춘 [그림 2]에서 다시 전송이 시작되려면 [그림 3]과 같이 수신 측에서 ACK를 보내야 합니다. [그림 3]의 ACK는 Host A가 보낸 0번 패킷에 대한 ACK이며, Host A는 0번 패킷에 대한 ACK를 받았으므로 현재 Host A가 ACK 없이 보낸 패킷은 1~4까지 4개의 패킷입니다. 하지만 5로 설정된 윈도우 사이즈에 따라서 Host A는 ACK 없이 5개 까지 패킷을 보낼 수 있으므로 0번 패킷에 대응하는 ACK를 받자마자 5번 패킷을 송신합니다. 이로써 ACK 없이 전송된 패킷은 시퀀스 번호 1에서 5까지 5개의 패킷이며, 윈도우 사이즈와 같은 수의 패킷이 ACK 없이 보내진 상황이므로 추가의 ACK 메시지가 오지 않으면 전송은 멈춘 상태로 지속됩니다.

이는 [그림 3]에서 Host A 위에 표시된 것과 같이 패킷을 나타내는 블록들 위로 녹색 사각형으로 표시한 슬라이딩 윈도우가 시퀀스 번호 1~5의 패킷을 포함하도록 한 칸 우측으로 이동한 것과 같습니다. 즉 슬라이딩 윈도우 안에 포함된 패킷들은 ACK 없이 일단 보내진 패킷들이고, ACK를 하나씩 받을 때마다 패킷을 하나씩 더 보낼 수 있으므로 슬라이딩 윈도우를 이동시키면서 현재 보내진 패킷들을 식별할 수 있습니다. [그림 3]에서 Host A 위의 0번 패킷을 회색으로 표시한 것은 ACK 수신으로 인하여 전송이 확인된 것을 나타낸 것입니다.

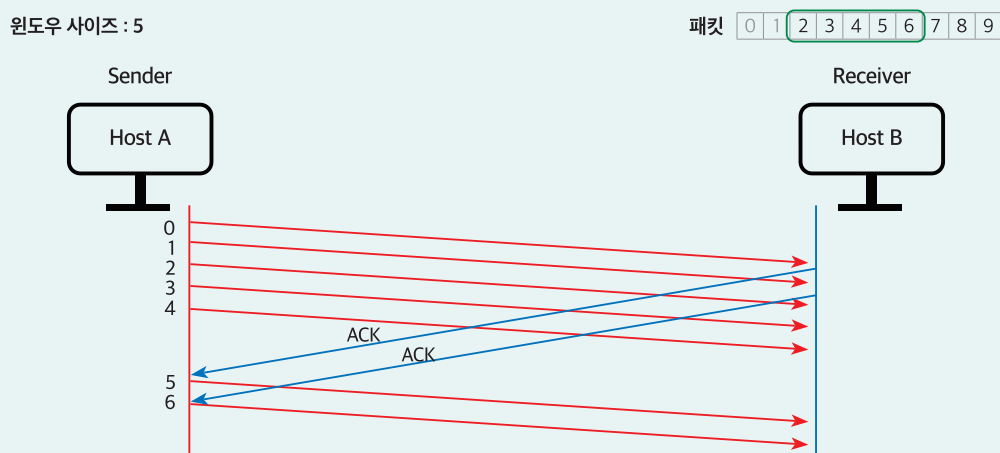


그림 4. 슬라이딩 윈도우 사이즈에 따른 패킷 전송 (ACK 메시지 추가 수신 후)

슬라이딩 윈도우 방식에 대한 익숙한 이해를 위해 0번 패킷에 대한 ACK를 수신하고 5번 패킷을 전송하여 1~5번까지 5개의 패킷이 ACK 없이 전송된 [그림 3] 이후의 상황을 한 번 더 가정해보겠습니다. [그림 3]에서 추가적 전송이 진행되려면 [그림 4]와 같이 수신 측인 Host B에서 1번 패킷에 대한 ACK를 Host A로 보내야 합니다. Host B가 1번 패킷에 대한 ACK를 보내면 Host A는 현재 2~5까지 4개 패킷을 ACK 없이 보낸 상태가 되고 5로 설정된 윈도우 사이즈에 따라서 Host A는 ACK 없이 5개까지 패킷을 보낼 수 있으므로 6번 패킷을 송신합니다. 이는 [그림 4]에서 Host A 위 녹색 사각형으로 표시된 슬라이딩 윈도우가 시퀀스 번호 2~6의 패킷을 포함하도록 [그림 3]에서보다 한 칸 우측으로 이동한 것과 같습니다. [그림 4]에서 Host A 위의 0번과 1번 패킷은 ACK 수신으로 인하여 전송이 확인된 패킷들임을 식별하기 위해 회색으로 표시했습니다.

지금까지 슬라이딩 윈도우에 대해 설명해 드렸습니다. 슬라이딩 윈도우는 Stop & Wait가 더 많은 데이터 전송이 가능한 상황에서도 ACK를 기다리며 전송을 하지 못하는 문제를 해결하기 위해 고안되었고 쉬운 이해를 위해 윈도우 사이즈만큼 패킷들을 보내고 ACK를 기다리는 상황 위주로 설명해 드리다 보니 최초에만 여러 개의 패킷을 보내고 이후는 Stop & Wait와 별 차이가 없는 것으로 보일 수 있으나, 패킷 전송과 ACK 수신에 동시에 일어나는 상황에서는 불규칙한 ACK 메시지 수신 타이밍을 윈도우 사이즈가 완화하는 작용을 하여 Stop & Wait보다 원활하게 패킷을 보낼 수 있습니다.

지난 편부터 지금까지 설명한 Stop & Wait, Stop & Wait ARQ 및 슬라이딩 윈도우 등은 어쩌면 Go-Back-N ARQ 및 Selective Repeat ARQ를 이해하기 위한 디딤돌에 지나지 않을 수도 있다는 생각이 듭니다. 슬라이딩 윈도우를 이용해 전송 효율을 높인 Go-Back-N ARQ 및 Selective Repeat ARQ가 현재 TCP에서 주로 사용되는 전송 관련 제어방식이기 때문입니다. 그럼 다음 편에서 Go-Back-N ARQ 및 Selective Repeat ARQ로 찾아뵙겠습니다. 📖

P.S.

C군이 여러분께 전하는 내용 중 전문적이라 볼 수 있는 분야의 지식은 대부분 독학으로 주경야독하여 전해드리는 것들입니다. 전문적 내용을 엄밀한 언어를 사용하여 전달하는 데 한계가 있으므로 설명하는 대상에 대한 대략적 이해에만 이용하시고, 그 이상은 권위 있는 전문자료를 참고하시기 바랍니다.
